CEWES MSRC/PET TR/98-16

Visualization of Damaged Structures

by

M. Pauline Baker
Dave Bock
Randy Heiland
Michael M. Stephens

04h00498

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC 94-96-C0002
Nichols Research Corporation

Visualization of Damaged Structures

M. Pauline Baker, Dave Bock, Randy Heiland
NCSA, University of Illinois
[baker | dbock | heiland]@ncsa.uiuc.edu

Michael M. Stephens
Army High-Performance Computing Research Center
stephens@wes.army.mil

March 1998

Abstract: Researchers working at high-performance computing centers are producing ever-larger data sets. Visualization of this data is crucial to understanding the phenomenon under study. But the sheer size of the data implies a significant set of challenges in data management and visualization. This report summarizes our experience in working with a very large-scale shock physics simulation.

1. Introduction

Easy access to large-memory, high-performance compute machines with sizable disk space has enabled researchers to tackle increasingly large problems. These computations produce data that can run into many Gigabytes. In this report, we summarize our experience in working with data from a large-scale shock physics calculation. The computation is part of a Department of Defense Challenge Project and was done at the U.S. Army Corps of Engineers Waterways Experiment Station (CEWES) Major Shared Resource Center (MSRC). Visualization specialists from the National Center for Supercomputing Applications (NCSA) collaborated with CEWES MSRC visualization personnel to visualize this data. The project was carried out under the Programming Environment and Training (PET) Initiative within the DoD's High-Performance Computing Modernization Program.

The goals of the visualization effort were to (1) provide support for this particular Challenge application, (2) highlight the science by showing the visualization at Supercomputing 97, and (3) work towards advancing the methodology for managing and interpreting data from very large-scale calculations. The intent to show the application at Supercomputing 97 meant there was an inflexible deadline for some of our activities. Additionally, tackling new computational challenges, particularly of this complexity and size, always includes unexpected problems that delay the progress of the research. So as with many visualization projects, problems with data availability and external time constraints limited what we could do in relation to the our second goal.

2. Simulation and Data Management

The Challenge application aims to simulate a bomb blast and its impact on a neighboring building. Two computational codes are used. CTH is used to simulate the blast propagation. Results of the CTH calculation are used as initial pressure loadings on the buildings and Dyna3D is then used to model the structural response of the building to the blast.

CTH, developed at Sandia National Laboratories, is a multi-material, large deformation, strong shock wave, solid mechanics code. It has models for multi-phase, elastic-viscoplastic, porous and explosive materials. The computational domain of the CTH problem was a 3-dimensional rectilinear grid with unit spacing on each of the *x*, *y*, and *z* dimensions. Within this computational space, cells are populated with different materials for the building, air, and the bomb.

Dyna3D is a nonlinear structural dynamics code available from Lawrence Livermore National Laboratory. It is based on finite element analysis techniques and can simulate the effect of stress on structures. The computational domain of the Dyna3D problem was a finite element mesh consisting of quadrilateral thin-shell elements that formed the entire building. Figure 1 shows the outer shell of the building (along with an isosurface and a slice from the blast calculation pressure field). The interior of the building was modeled with considerable detail, including rooms, doorways, and windows, also shown in Figure 1.

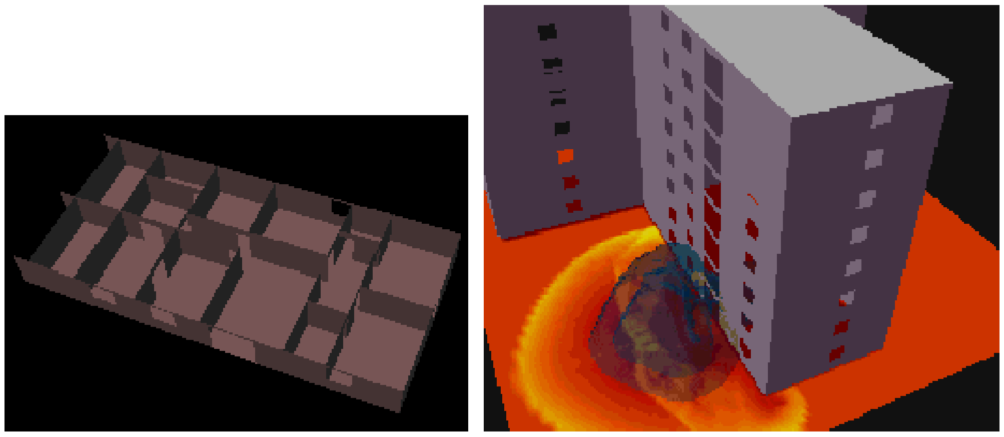


Figure 1 . The building was modeled with 270,000 thin-shell elements. This figure shows the outer hull of the building on the right. The interior of the model included walls, windows, and doorways.

The computational grid for the CTH run was 460 x 516 x 312 (about 74M cells. The simulation executed on 256 processors of the T3E. *Restart* files were saved periodically, with information on over 35 variables. *Plot* files were extracted from the restart files, saving a subset of the variables, including pressure, velocity, mass, and cell volume. There were 155 timesteps saved as plot files, representing 15.5 ms of simulated bomb blast. Plot files were saved on a per-processor basis, so there were almost 40,000 data files for the CTH simulation. Also saved were complete time histories, including information for all the variables, for about 200 physical locations. These were placed at circular locations surrounding the initial placement of the bomb.

CTH also provides an alternate file format, commonly referred to as *vis* files. This format would have been easier to handle and would have facilitated subsequent visualization efforts. However, early tests showed that *vis* files worked for small data sets, but failed when applied to a computational grid of this size.The Dyna3D simulation was run on the MSRC’s C90. There were 164 timesteps, representing 150 ms of simulated time. For each timestep, a Dyna *state file* was produced containing data for 45 variables, including information on the changing geometry of the building elements. Of the other variables, the researchers were most interested in *effective plastic strain*, which can serve as an indicator of damage to a building component. Beyond a certain plastic strain threshold, the structural element is assumed to have failed.

3. Visualization Goals

For the bomb blast simulation, the researchers were particularly interested in seeing the progress of the blast's shock front as it hit and went over the top of the building, so the pressure field from the CTH data was the most important component. There was also considerable interest in looking at the velocity of the shock front. We concluded that we would want to look at an isosurface of the pressure field, at 1.4e05 pascals, and that it might also be useful to see this isosurface colored by velocity. We were also interested in representing the multivariate time histories saved for the 200 points scattered through the bomb blast space.

For the Dyna3D component of the problem, our researcher team needed visualization both for monitoring the progress of the run as well as exploring the data in a post-processing mode. Figure 2 shows selected timesteps from an early Dyna3D run, using a simplified model of the building with only 27,000 thin-shell elements. The visualization makes it abundantly clear that the run went astray after a certain point.

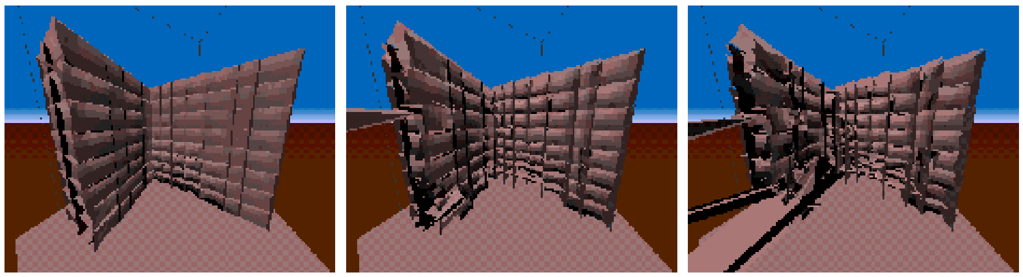


Figure 2. An early run of the Dyna3D simulation, using a simplified building model. The visualization makes it clear that the run went astray.

In a post-processing mode, we wanted to support examination of the structural deformation from a variety of viewpoints. Bomb blast damage can be asymmetric -- with adjacent areas receiving more or less impact from a bomb. Thus it seemed useful to be able to position the observer at various viewpoints and examine the propagation of the shock front and the subsequent building response. Also of interest was the ability to analyze the building deformation with the shell elements colored by the plastic strain parameter.

4. Visualization Tools and Techniques

Given the goals as described above, a variety of well-understood visual representations could be applied, such as isosurfaces and colored slices for the CTH pressure data. We had some ideas about how to represent and explore the multivariate time histories. But we also anticipated that the greatest challenges in this project would come from the sheer size of the data, particularly since we wanted to provide for the researchers the capability to interactively explore the data.

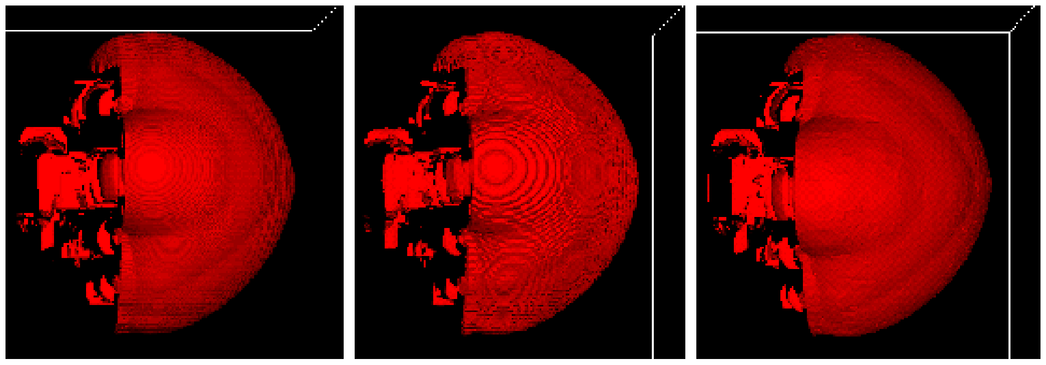
The second source of greatest challenge came from the calendar and the clock. The computational challenges of this project were significant. Although we had some preliminary data, the full data sets were not available until very close to Supercomputing 97. Still we were committed to highlighting the science by showing the results at that meeting.

We followed a strategy of subsampling the CTH data to make it more manageable, mapping that data to geometric form, and supporting interactive exploration of the playback of that geometry. Similarly, for the Dyna3D data, we extracted out the region of highest interest, decimated the geometry where possible, and supported interactive playback of the geometry. We used AVS to experiment with visual representations and colormaps, and we used VTK (the Visualization ToolKit) to extract the final geometries. The interactive playback tool was constructed from SGI's Performer library with a Motif interface.

As described above, the CTH data was saved as per-processor, per-timestep plot files. The CTH package includes a utility program, cthed, to extract single variables from the plot files. We used a collection of perl and expect scripts to (1) gather the 256 files containing data for a timestep, (2) use cthed to extract pressure, yielding an ASCII file of about 3.2 MB, (3) knit the 256 separate files together, subsampling by 2 in each dimension along the way to produce data blocks that were 230 x 258 x 156.

We took advantage of the fact that there was a particular isosurface value of interest: 1.4e05 pascals pressure. Interactive generation of a variety of isosurface values would not be necessary. Rather we could extract that isosurface from the data for each timestep in advance and work only with the resulting geometry rather than the data itself. While the isosurface files were certainly smaller than the data, our desire for an interactive application raised concerns about the size of the geometry files and the number of triangles required to draw an isosurface.

We experimented with again subsampling the data before extracting the isosurface. Alternatively, we used VTK's decimation features to reduce the number of polygons from the higher-resolution data. Figure 3 compares the results. The original isosurface shown in (a) contains 306239 triangles. The middle surface (b) was extracted from data that has been subsampled by 2. The resulting shape has only 72751 triangles, but is of unacceptable quality. Finally, the isosurface in (c) was produced by decimating the original isosurface (a). This final isosurface contains only 62532 triangles. Decimation was judged to be the better route to take, so the isosurfaces were computed on the 230 x 258 x 156 data blocks and then decimated.



(a) (b) (c)

Figure 3. (a) An isosurface from the higher-resolution pressure field, with 306239 triangles. (b) Subsampling by 2 in each dimension yields an isosurface of only 72751 triangles, but of obviously inferior appearance. (c) Decimating the isosurface from (a) produces a satisfactory isosurface of only 62532 triangles.

To provide more context for the isosurface, we also extracted and colored (again, using VTK) a collection of slicing planes from the pressure data. We extracted slices where i = 124, i = 160, and a "ground" slice k = 20, and used a fire palette (dark crimson, orange, yellow, white) to color the slices. Figure 4 shows 3 selected timesteps of the k slice, with a faint outline of the building footprint. In addition to providing context for the isosurface (see Figure 1) the slices reveal the complexity of the development of the pressure field at values less than 1.4e05 pascals.

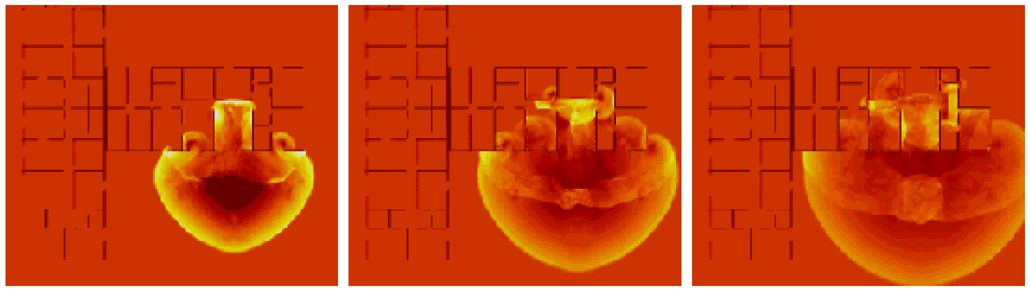


Figure 4. A fire palette was used in coloring the slices extracted from the pressure field. This slice is parallel to the ground, at k = 20.

The full building used in the Dyna3D simulation required 540,000 triangles to draw. Reaching our goal for an interactive application demanded that we reduce this load. The research team suggested that we exclude all but the "most interesting" part of the building, the lower floors on the wing directly impacted by the blast. VTK again proved useful in extracting out this geometry. Figure 5 shows two snapshots of this part of the building, at early and late moments in the Dyna run. The two slices (one on the ground and one vertical) are included as context only and are not necessarily timed correctly with the building deformation sequence.

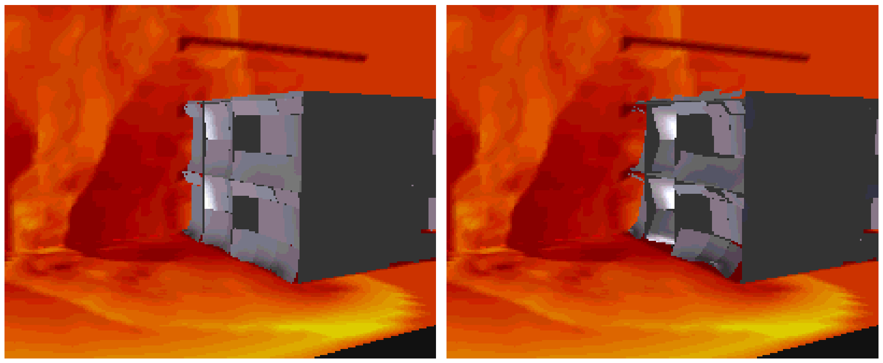


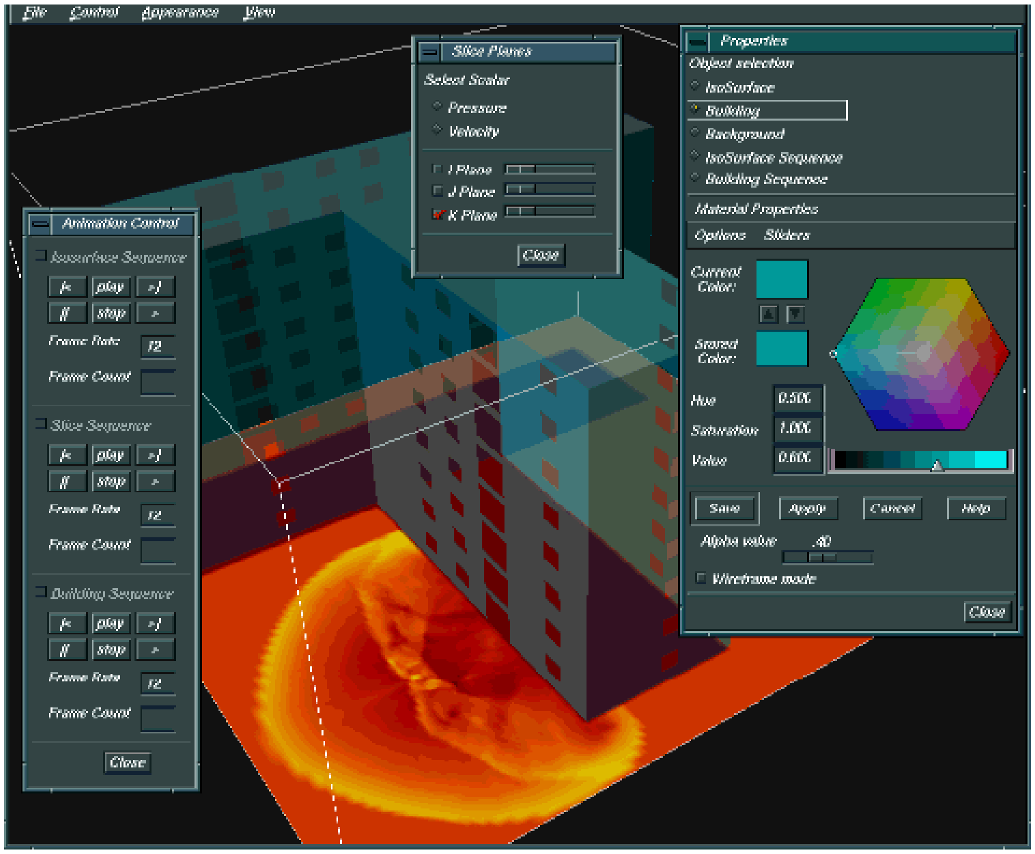
Figure 5. The lower 2 floors of the building on one wing, at an early and late moments in the sequence.

5. User Interface

The primary goal of this project was to support the DoD Challenge application in Damaged Structures. Providing the capability for interactive exploration of the data was a desired end. Given the time constraints imposed by data availability and the Supercomputing 97 meeting, we were limited to providing a tool that at least supported interactive playback of the pre-computed geometry. We constructed a demonstration application that allowed the researchers to exercise options for looking at the geometries. For example, they could vary the view position, or view the representations in different combinations.

The particular slices that had been precomputed were chosen without much conviction that these would be the only interesting slices. To expand the range of choices, we included in the interactive tool the capability to extract and color any slice from a single pressure time step. The entire pressure field is too large to be downloaded onto the typical (or even high-powered) visualization workstation. While we couldn’t support animating the whole time series at arbitrary slice planes, it seemed reasonable to support full exploration of at least a single timestep. Figure 6 shows a screen capture of the application.

This application was constructed as a custom viewer for the Damaged Structures Challenge. We have also found it applicable to a couple of other applications, simply by substituting the precomputed geometries that are to be displayed. Thus there have been unanticipated benefits.



Frame Rate 12

Hue 0.50
Saturation 1.00
Value 0.60
Alpha value .40

Figure 6. The finished application included a Motif user interface supporting user control over viewing parameters.

6. Alternative Strategies for Visualizing Large-Data Simulations

The strategies adopted in this effort included subsampling the CTH data, precomputing geometries for the isosurface value of interest and for a particular set of slices, and limiting the building geometry to either the outer hull or the lower floors in the region of interest. This procedure was adequate for the needs of this project but far better solutions are possible for dealing with very large data sets. In this section, we outline other possibilities for supporting visual analysis of data from large-scale simulations.

1. Coprocessing Systems and Parallel Visualization.

Coprocessing systems provide support for visualization to be performed concurrently with the computation, usually as a separate process, sometimes with the data in a buffer that is shared between computation and visualization. Mapping data to visual form is carred out and can be captured during the course of the run, rather than as a post-processing step. Thus coprocessing systems support computational monitoring and interactive steering. Embedding CTH within a coprocessing system would allow the user to monitor the progress of the run, perhaps stopping a run early if it appears that the run will not yield useful results.

If visualization is to be done on high-performance compute machines, possibly in the context of a coprocessing system, parallel implementations of the visualization algorithms involved are necessary to insure responsible use of resources.

2. **Multiresolution Representations.**

   Multiresolution representations provide another approach to handling very-large data simulations. Providing representations at various levels of detail allows the user to interact with coarser representations while still retaining access to higher-level resolution representations.

3. **Image-based Rendering.**

In some cases, it is desirable to dispense with the geometry altogether and rely on imagery for showing the data to the researcher. A variety of efforts are underway in the visualization community to experiment with algorithms and techniques to utilize image-based visualization effectively. Developments in the hardware arena, such as SGI's support for including InfiniteReality graphics boards in their high-performance Origin2000 machine, lend additional support for image-based visualization techniques.

7. Summary

In this effort, we adopted 3 goals: (1) to provide support for this particular Challenge application in large-scale shock physics and residual damage modeling, (2) to highlight the science by showing the visualization at Supercomputing 97, and (3) to work towards advancing the methodology for managing and interpreting data from very large-scale calculations. For the Supercomputing 97 meeting, we produced a visualization tool that showed the bomb blast propagation and the building deformation. Time constraints forced us to rely on time-tested techniques. We followed a strategy of subsampling the CTH data to make it more manageable, mapping that data to geometric form, and supporting interactive exploration of the playback of that geometry. Similarly, for the Dyna3D data, we extracted out the region of highest interest, decimated the geometry where possible, and supported interactive playback of the geometry. We have subsequently reevaluated those procedures and made plans for follow-on work in supporting visualization of large-data simulations.